0[9] We can define bunch \textit{powint} by the equation \textit{powint} = 2^{\textit{int}}. Define the same bunch by construction and induction, without using exponentiation.

1 Let \textit{i} be an integer variable, and let \textit{P} be a specification such that
\begin{align*}
P & = \text{if } i=0 \text{ then } \text{ok } \text{ else } i := i-1. \text{ P. } i := i+1 \text{ fi} \\
(\text{a}) & \text{ Add recursive time.} \\
(\text{b}) & \text{ Using recursive construction starting with } \top, \text{ find the first three approximations toward the weakest fixed-point (weakest solution for } P), \text{ including time.} \\
(\text{c}) & \text{ From these three approximations, guess the weakest fixed-point (weakest solution for } P). \\
\end{align*}

2[12] The axioms of program Queue Theory are
\begin{enumerate}
\item \textit{isemptyq}’ \iff \textit{mkemptyq} \\
\item \textit{isemptyq} \Rightarrow \textit{front}’ = x \land \neg \textit{isemptyq}’ \iff \textit{join x} \\
\item \neg \textit{isemptyq} \Rightarrow \textit{front}’ = \textit{front} \land \neg \textit{isemptyq}’ \iff \textit{join x} \\
\item \textit{isemptyq} \Rightarrow (\textit{join x. leave} = \textit{mkemptyq}) \\
\item \neg \textit{isemptyq} \Rightarrow (\textit{join x. leave} = \textit{leave. join x})
\end{enumerate}
Prove \textit{front}’=3 \iff \textit{mkempty. join 3}